

```
entry:
%stktop_8 = alloca i8, i32 24, align 1
%tos = ptrtoint ptr %stktop_8 to i64
%0 = add i64 %tos, 16
%RBP_N.8 = inttoptr i64 %0 to ptr
%1 = add i64 %tos, 0
%RSP_P.0 = inttoptr i64 %1 to ptr
store i64 3735928559, ptr %RSP_P.0, align 8
%RBP = ptrtoint ptr %RSP_P.0 to i64
store i64 %arg1, ptr %RBP_N.8, align 1
%2 = ptrtoint ptr %stktop_8 to i64
%3 = add i64 %2, 4
%4 = inttoptr i64 %3 to ptr
store i32 0, ptr %4, align 1
br label %bb.1
```



```
bb.1:
%5 = ptrtoint ptr %stktop_8 to i64
%6 = add i64 %5, 4
%7 = inttoptr i64 %6 to ptr
%memload = load i64, ptr %7, align 1
%8 = trunc i64 %memload to i32
%RAX = sext i32 %8 to i64
%9 = sub i64 %RAX, 8
%10 = call { i64, i1 } @llvm.usub.with.overflow.i64(i64 %RAX, i64 8)
%CF = extractvalue { i64, i1 } %10, 1
%ZF = icmp eq i64 %9, 0
%highbit = and i64 -9223372036854775808, %9
%SF = icmp ne i64 %highbit, 0
%11 = call { i64, i1 } @llvm.ssub.with.overflow.i64(i64 %RAX, i64 8)
%OF = extractvalue { i64, i1 } %11, 1
%12 = and i64 %9, 255
%13 = call i64 @llvm.ctpop.i64(i64 %12)
%14 = and i64 %13, 1
%PF = icmp eq i64 %14, 0
%CFCmp_JAE = icmp eq i1 %CF, false
br i1 %CFCmp_JAE, label %bb.3, label %bb.2
```

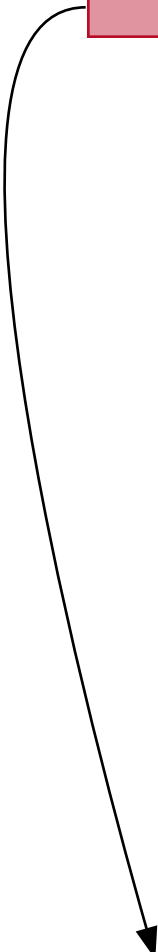
T

F



```
bb.2:
%15 = ptrtoint ptr %stktop_8 to i64
%16 = add i64 %15, 4
%17 = inttoptr i64 %16 to ptr
%memload1 = load i32, ptr %17, align 1
%EAX = call i32 (ptr, ...) @printf(ptr getelementptr inbounds ([112 x i8],
... ptr @rodata_15, i32 0, i32 16), i32 %memload1)
%memload2 = load i64, ptr %RBP_N.8, align 1
%18 = ptrtoint ptr %stktop_8 to i64
%19 = add i64 %18, 4
%20 = inttoptr i64 %19 to ptr
%memload3 = load i64, ptr %20, align 1
%21 = trunc i64 %memload3 to i32
%RDX = sext i32 %21 to i64
%memref-basereg = add i64 %memload2, %RDX
%22 = inttoptr i64 %memref-basereg to ptr
%memload4 = load i32, ptr %22, align 1
%23 = trunc i32 %memload4 to i8
%ESI = sext i8 %23 to i32
store i32 %EAX, ptr %stktop_8, align 1
%EAX5 = call i32 (ptr, ...) @printf(ptr getelementptr inbounds ([112 x i8],
... ptr @rodata_15, i32 0, i32 19), i32 %ESI, i64 %RDX, i64 %memload2)
%24 = ptrtoint ptr %stktop_8 to i64
%25 = add i64 %24, 4
%26 = inttoptr i64 %25 to ptr
%memload6 = load i32, ptr %26, align 1
%EAX13 = add i32 %memload6, 1
%27 = call { i32, i1 } @llvm.uadd.with.overflow.i32(i32 %memload6, i32 1)
%CF7 = extractvalue { i32, i1 } %27, 1
%28 = and i32 %EAX13, 255
%29 = call i32 @llvm.ctpop.i32(i32 %28)
%30 = and i32 %29, 1
%PF8 = icmp eq i32 %30, 0
%ZF9 = icmp eq i32 %EAX13, 0
%highbit10 = and i32 -2147483648, %EAX13
%SF11 = icmp ne i32 %highbit10, 0
%31 = call { i32, i1 } @llvm.sadd.with.overflow.i32(i32 %memload6, i32 1)
%OF12 = extractvalue { i32, i1 } %31, 1
%32 = ptrtoint ptr %stktop_8 to i64
%33 = add i64 %32, 4
%34 = inttoptr i64 %33 to ptr
store i32 %EAX13, ptr %34, align 1
br label %bb.1
```

```
bb.3:
ret i64 %RAX
```



CFG for 'print\_char' function